Sustainable Sprint

SUSTAINABLE SOFTWARE ENGINEERING

CS4575

Authors:

Wout Burgers (5329868)
Martijn Frericks (5511542)
Thomas Rooskens (5554950)
David van der Maas (5083346)

April 4, 2025

Delft
University of
Technology

Abstract

As software becomes increasingly important to life, its impact on the environment remains invisible. The growth of demand for computer power and internet services is a massive contributor to global energy demand. The majority of developers are unaware that the ineffective use of resources by coding leads to unnecessary waste. Traditional educational methods such as textbooks and lectures in classrooms do not engage enough with the learner and provide a hands-on experience in acquiring sustainable software engineering. In order to bridge this gap, this report presents **Sustainable Sprint**, an educational board game derived from "The Game of Goose." This interactive educational tool uses knowledge based, puzzle solving and elements of chance to educate players on sustainable software development principles in an entertaining manner. By promoting this awareness and adopting best practices in green software engineering, **Sustainable Sprint** aims to equip the developers with the knowledge necessary to create energy-efficient and eco-friendly software.

I. Introduction

Software is a part of modern society, but its environmental impact is not considered seriously. The increasing demand for computing power and internet services contributes a significant amount of energy demand globally [12]. However, most developers, especially new developers, are unaware that poor coding methods may lead to wasteful usage of resources. Traditional educational methods, such as lectures and textbooks, may not be able to motivate students or provide practical learning in green software engineering.

To fill this gap in knowledge, we propose a new educational tool: a board game called Sustainable Sprint based on "The Game of Goose." Using interactive mechanics involving knowledge-based puzzle-solving with randomness, this board game educates primary sustainability concepts in software engineering in a fun and experiential way. By responding to questions about sustainability and gaining exposure to real-world coding scenarios, players will become familiar with energy-efficient coding techniques. Designed for new programmers and experienced professionals, the game provides an entertaining and competitive learning experience with sensitivity to green coding.

To ensure that the game is fun as well as educational we have tested the game extensively with participants. We observed with the playtests how the players interacted with the game mechanics, learned through their knowledge of sustainability concepts. With this we obtained feedback regarding the balance, difficulty and fairness of the game. This feedback helped us to make several changes to the game including adjusting question difficulty, explanation of rules and the distribution of game tiles to ensure a smoother and more enjoyable game.

Furthermore, to ensure maximum accessibility of the game and encourage the widespread adoption of it, we are publishing the game for free as an open-source project on GitHub[1]. This will allow teachers, developers and other people to download and use the game for various learning environments, such as classrooms, workshops and professional training. By making it a lot of fun and accessible, our board game will encourage a new generation of programmers to adopt these more eco-friendly coding practices and build a power efficient online world.

II. Problem Statement

As computing energy usage continues to grow sustainability in software engineering has become a critical concern. There are many programmers that lack awareness of how poor coding contributes to inefficient energy use and resource wastage. Existing educational methods fail to provide interactive and engaging ways to teach sustainable coding. Additionally, most educational games focus on general programming skills without addressing the environmental impact of software development.

Traditional teaching methods, such as textbooks and lectures, are not likely to motivate students and lack practical experience in applying sustainability concepts. In addition, existing educational games focus on general programming skills without emphasizing the environmental impact of software development.

Therefore a new approach is needed to make learning about sustainable coding enjoyable, intuitive and suitable for students with varying programming experience. Our board game aims to fill this gap by introducing the key sustainability concepts through this interactive gameplay. Players will answer sustainability related trivia and tackle real challenges, making learning both fun and practical.

III. Background and Related Work

This section reviews the existing work on gamification in education and its applications in sustainable software engineering, highlighting relevant games that promote energy efficiency and sustainable decision-making.

A. Gamification

Gamification is an innovative approach to education, offering an engaging way to help students learn [2]. By incorporating game-like elements such as points and rewards, students are encouraged

[1] https://github.com/Thomsr/sustainable-sprint

to actively participate in their learning. This not only makes the process more enjoyable but also provides a sense of achievement and progress as students tackle challenges or answer questions. Through these interactive experiences, learners can better understand and apply complex concepts. For example, participants of the game can learn concepts such as sustainable software engineering while staying motivated by the game-like structure.

One of the main advantages of gamification in education is its ability to turn passive learning into active participation [13]. Instead of simply reading textbooks or attending lectures, students are immersed in engaging activities that require them to think critically and solve problems. As they progress through levels or earn points for answering questions, they are motivated to continue learning and improving their skills. The use of rewards and milestones helps create a sense of accomplishment, making the learning process feel more like a game and less like a traditional classroom task.

Does Gamification work? According to a study conducted in 2014 [5], gamification produces mostly positive effects and benefits. The outcomes vary based on the nature of the game and the characteristics of the users. Some studies found that the effects on gamification on learning was only short-term, potentially driven by novelty. The study also highlighted the need for more studies to better understand when gamification is most effective.

B. Sustainable (software) games

In recent years, the importance of sustainable software engineering has grown, leading to the development of educational games on the topic. As mentioned earlier, these games combine learning with entertainment, helping players understand and apply best practices for sustainable software engineering [7].

One such game is Echoes of the Future, developed by students at the University of Amsterdam. The game revolves around climate change and sustainable software engineering and allows players to learn about the environmental impact of different technological decisions. Through storytelling and interactive play, it illustrates the importance of sustainable decision-making in software engineering and how digital technologies can contribute to a more sustainable future. There are other games, such as Green With Energy, which, while not very software engineering in nature, educates users on saving energy and optimizing its use. These games are designed to simulate energy issues where users have to design sustainable energy systems or optimize energy consumption to reduce waste. By mimicking the complexity of energy management, it emphasizes the importance of reducing wastage and optimizing resources.

IV. Solution Proposal

To promote sustainable software engineering in an engaging and interactive way, we propose a board game adaptation of "The Game of Goose". By introducing sustainability related challenges and decision-making scenarios, our game encourages players to develop a deeper understanding of environmentally responsible software development. This section outlines the core idea behind our game, its mechanics and how it adapts to different levels of programmers.

A. Idea

To solve this, we will modify a popular board game, “The Game of Goose”. The goal of the game is to be the first player to reach the last square. Players can advance along the board by rolling a die that determines the number of steps they can take. However, be careful, because the board contains special squares. Landing on one of these squares presents a card which may read the following: “You use inefficient algorithms with a high time complexity, go back 3 spaces” or other anti-patterns for engineering sustainable software.

We also want to reduce the luck factor of the game, e.g. asking people questions related to sustainability. If a question is answered correctly, a player can move forward x steps. If the answer is wrong, the player has to move x steps back.

B. Target Audience

We aim to develop a game that caters to both beginner and experienced programmers by incorporating two distinct sets of questions and statements. The first set is designed for individuals with zero to little programming experience, such as high school students or beginners. These questions and statements will focus on fundamental programming concepts, basic sustainable coding practices, and general awareness of software’s environmental impact. Examples of topics covered in this set include:

- The importance of writing clean and efficient code to reduce energy consumption.
- Examples of applications that consume a lot of energy, with real-life cases.
- How turning off unused processes can conserve energy.
- General coding best practices that support sustainability and efficiency.

The second set is tailored for experienced programmers, such as computer science students, software engineers, and IT professionals who already have a strong understanding of programming fundamentals. This advanced set will introduce complex, real-world scenarios that require players to apply their knowledge of sustainable software engineering in more technical and strategic ways. Topics in this set include:

- The impact of different programming languages on energy consumption.
- How data structures and algorithms can optimize software efficiency and reduce processing power.
- Strategies for writing low-power, high-performance applications.

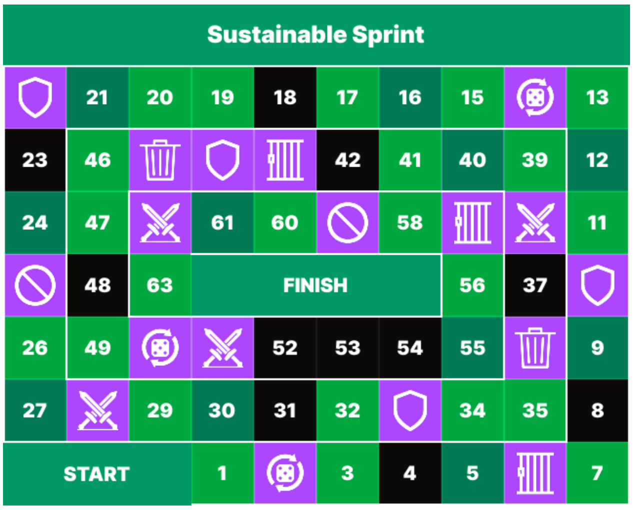
By implementing these two levels of questions and statements, the game ensures an engaging and educational experience for all players, regardless of their technical background. Beginners will gain a foundational understanding of sustainable software practices, while advanced players will be challenged to think critically about real-world industry issues and best practices for environmentally friendly coding.

V. The Game

In the following sections, we will introduce the game board and the different game elements included in our variant of "The Game of Goose", called "Sustainable Sprint".

A. Game Elements

The game board, as shown in Figure 1, is made up of various types of tiles that incorporate strategy, challenge, and interaction in the gameplay. Each tile offers players specific situations about sustainable software engineering or generic game mechanics. There are tiles that pose questions to players related to sustainability, while others have penalties for bad coding habits or accolades for sustainable choices. Also, event tiles bring an element of surprise and competition into the game, which makes it interesting and dynamic. The players must navigate these tiles tactically to reach the final square first while learning tips on green coding techniques along the way.



Sustainable Sprint

🛡	21	20	19	18	17	16	15	🎲	13
23	46	🗑	🛡	▥	42	41	40	39	12
24	47	⚔	61	60	🚫	58	▥	⚔	11
🚫	48	63	FINISH				56	37	🛡
26	49	🎲	⚔	52	53	54	55	🗑	9
27	⚔	29	30	31	32	🛡	34	35	8
START		1	🎲	3	4	5	▥	7	

Figure 1: The Sustainable Sprint game board

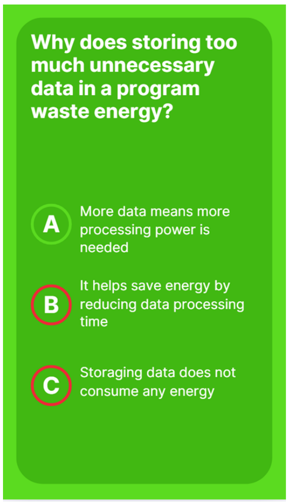
In the following subsections we will go over the different types of tiles that the game will include.

1) Green Energy Question (23 tiles)

The Green Energy Questions are designed to make learning about sustainable software practices fun and engaging. In this part of the game, players answer multiple-choice questions focused on green coding and software sustainability. A correct answer moves them forward two tiles, while an incorrect answer sends them back two tiles. This setup adds an element of challenge and rewards players for getting eco-friendly coding practices right.

We've created two sets of questions to accommodate different levels of expertise: a beginner set and an advanced set, as described in Section IV-B. An example for both questions can be seen in Figure 6 and Figure 7. This way, players of both skill levels can participate. The beginner questions introduce key concepts like how simplifying code can reduce energy consumption or how choosing the right programming language can make a difference. The advanced questions go deeper, exploring topics such as optimizing cloud computing or understanding the carbon footprint of software.

The questions are based on ideas on sustainable software design, drawing from sources that focus on reducing energy consumption in software development. The questions cover topics such software architectures like microservices and monolithic architecture and their energy consumption [1, 14], leveraging caching to reduce energy consumption [4, 16], picking energy-efficient data structures [10], energy-efficient cloud computing and network traffic [6, 8, 15], green software design [9, 11], and the energy cost of cybersecurity [3]. These give a global basis for the energy consumption of software and the principles of sustainable software engineering.



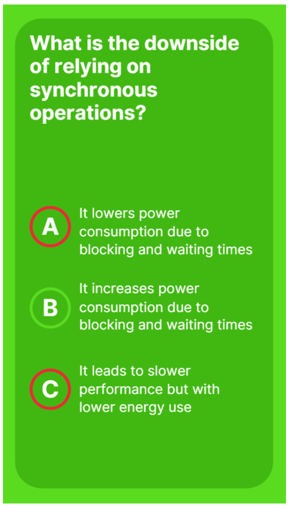
Why does storing too much unnecessary data in a program waste energy?

A — More data means more processing power is needed

B — It helps save energy by reducing data processing time

C — Storaging data does not consume any energy

Figure 2: An example question for beginner level users



What is the downside of relying on synchronous operations?

A — It lowers power consumption due to blocking and waiting times

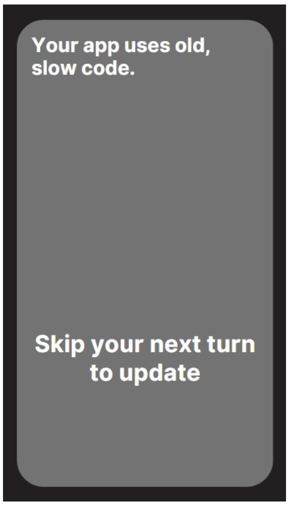
B — It increases power consumption due to blocking and waiting times

C — It leads to slower performance but with lower energy use

Figure 3: An example question for advanced level users

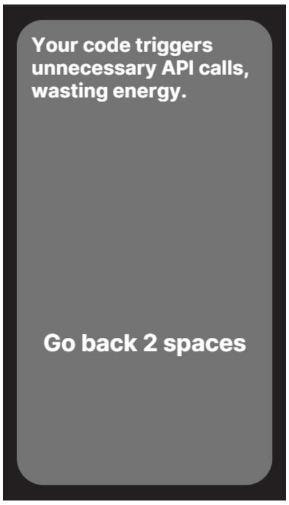
2) *Energy Black Hole (11 tiles)*

This tile is just a statement. The player will be shown bad software practices like unnecessary energy consumption. Players must skip a turn or move backwards.


Your app uses old, slow code.

Skip your next turn to update

Figure 4: An example black hole statement for beginner level users


Your code triggers unnecessary API calls, wasting energy.

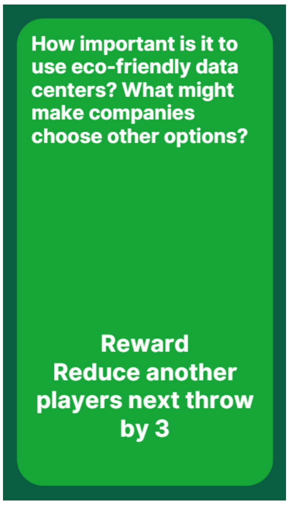
Go back 2 spaces

Figure 5: An example black hole statement for advanced level users

3) *Eco-Reward (11 tiles)*

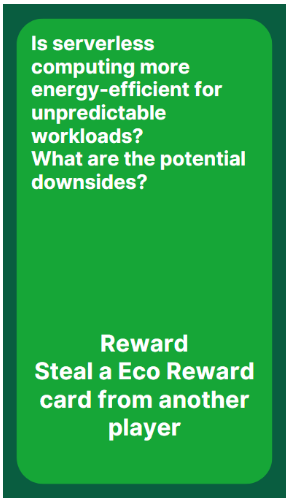
This tile will also only show a statement about good sustainable software practices. It will reward players for making sustainable decisions. These card can be saved for a later moment. Depending on the card, a player can:

- Double their dice throw.
- Go forward a couple of steps.
- Make another player move backwards.


How important is it to use eco-friendly data centers? What might make companies choose other options?

Reward
Reduce another players next throw by 3

Figure 6: An example eco reward question for beginner level users


Is serverless computing more energy-efficient for unpredictable workloads?
What are the potential downsides?

Reward
Steal a Eco Reward card from another player

Figure 7: An example eco reward question for advanced level users

4) *Event tiles (18 tiles)*

These tiles have nothing to do with sustainable software engineering, but are created for more engagement. These tiles will include:

- **Prison (3)**: You can only continue playing when you throw a 6.
- **Roll again (3)**: The player may roll again.
- **The next player skips a turn (2)**: The next player loses their turn.
- **Safe zone (4)**: Nothing happens, you can relax for a turn.
- **Dice battle (4)**: You can choose a player whose place you would like to change with. You both throw the dice. Did you throw the highest, then you change places. Did you throw the same amount or lower? Then nothing happens.
- **Lose your event cards (2)**: Do you have any event cards stacked up? Then you lose all of them.

VI. Play Testing

To determine the effectiveness and playability of our game, we conducted playtesting with some participants who were novice programmers. We were interested in learning how quickly the players could learn the game mechanics and sustainability principles without any introductory guidance.

The players were provided with the rulebook and game board without additional instructions. We allowed them to find out and play the game independently, letting them interact with the game naturally. During the session, we observed their decision-making, interaction with different types of tiles, and reactions to sustainability issues.

Following the gameplay, we conducted a feedback session where participants shared their experiences, highlighting what they enjoyed and challenging aspects they encountered. The key areas of discussion were clarity of instructions, luck and strategy balance, and the educational applicability of sustainability principles in the game.

Overall, the playtest provided valuable feedback on what needs to be done differently, which will be addressed in the Validation and Improvements chapter.

VII. Validation

The playtesting allowed us to see how the game was played and how people were feeling while and after playing the game. While playing, we have looked at where the players had difficulties or did not like certain aspects. In the following sections, we will go over the feedback we received and the results of the short interview we conducted.

A. Improvements Made

The playtesting process also brought out some areas of improvement that have been added to sharpen the game mechanics and rule simplicity. One such major issue found was the evaluation of answers for the

Eco-Reward cards. The players at first responded to a question related to sustainability, and the team collectively determined if the answer was correct. This method caused imbalance, so we re-drafted the rule such that the Eco-Reward card is awarded to the player with the best response, so the game would be fairer and more competitive.

Another area of confusion was movement when players stepped onto tiles that instructed them to move backward or forward. A few participants were unsure if they should do the actions when moving onto a tile. To make gameplay more efficient, we established that only event tiles need an action when stepped on, while all other tiles when moving do not, to avoid unnecessary slowdowns and maintain the smooth game pace.

Lastly, the balance of question types was commented upon. Too many discussion questions were complained about by some players, as they made the game drag. Therefore, the layout of the tile types on the game board was adjusted. The players suggested that the number of Green Energy Question tiles be expanded and reduce the number of Energy Black Hole and Eco-Reward tiles. This change is to enhance the balance so that the proportion of challenge and reward will harmonize well with the quantity of question cards present. These changes will enhance the entire game, making it more interactive, informative, and inclusive for all.

B. Interview Results

We posed some direct questions in the feedback session to assess participants' experience. We have summarized the responses and included the useful feedback:

How did you experience the level of difficulty of questions? Most players found the expert questions to be fairly challenging, thus forcing them to contemplate the concepts of sustainability deeply. Although they enjoyed the thoughtfulness of questions, we decided to include some instructions to be able to choose which set of playing cards suit the players the best.

How did you experience the distribution of tiles? Players felt that the tile distribution could be improved. They suggested increasing the number of Green Energy Question tiles but decreasing the number of Energy Black Hole and Eco-Reward tiles. This would give a more balanced challenge and reward.

How did you experience fairness in terms of rewards and balance? Some players mentioned that the reward of a double roll of dice could give a tremendous advantage to the players who land on those squares more often. Even though this is a component of chance in board games, we made some minor adjustments to only have three roll again tiles.

What did you think of the event tiles? There were diverse opinions regarding the event tiles. While some appreciated them as thrilling and amusing, others felt that certain event tiles, like the dice battle could be improved in terms of balance.

Did you learn something? All participants indicated they learned at least something. They became more aware of energy-efficient coding practices and environmentally friendly software engineering concepts after participating in the game. The game's interactive features helped to drive home key points in an engaging way.

Would you play it again? In general, the majority of participants were enthusiastic about playing the game again. The competition, learning, and interactivity combined to make the game fun, and many suggested that it be used in workshops or classrooms at school as a fun yet useful way to introduce ideas of sustainability. However, it was mentioned that the chances of playing again would be higher with a new set of cards to make sure the same questions were not posed again.

VIII. Sharing the Game With The World

To ensure our game reaches the masses and is used as a learning tool all over the world, we have created a dedicated GitHub page[2] where all the components of the game, including the report, game board, cards, and rulebook, will be made free to download. This will allow teachers, sustainability activists, coders, and gamers to download and use the game in schools, workshops, businesses, and even on game nights.

By going open-source, we welcome community contributions where users can propose improvement, new questions, or even altering the game mechanics to accommodate other target audiences. The GitHub repository shall serve as a meeting point of commonality for discussion, updates, and further development, with an encouragement for a dynamic and participatory approach in sustainability education through games.

To also make it accessible, we will also provide print-friendly copies of the game board and cards so that anyone can easily make a physical version. We will also provide clear instructions on how to set up and play the game so that new users can easily learn and begin using it without prior experience. Our goal is to offer interactive, engaging, and accessible sustainable software engineering education that

[2] https://github.com/Thomsr/sustainable-sprint

promotes green coding practice awareness in an enjoyable way.

IX. Discussion

The playtesting session of the game has provided some precious feedback on the educational and play value of the game. Overall, the game was discovered to be an effective tool for introducing programmers to the domain of sustainable software engineering in a fun and stimulating way. It was observed from the feedback that players, be they experienced or inexperienced, liked the interactive blend of learning and competition. The majority of the participants stated that they gained more knowledge concerning effective coding and green coding practices, showing that the game was indeed able to fill a void in current approaches of green coding instruction.

One of the major challenges we had was achieving the ideal ratio of luck to strategy, as is typical in most board games. Whereas others liked the unpredictability that the event tiles and rewards added, some opined that sometimes it disturbed the flow of learning. That was a valid criticism only if it served to highlight the requirement that one needs to ensure the learning elements should override gameplay and that random events shouldn't override them. Therefore, adjustments to event tiles, and adjusting the distribution of question and reward types, will assist in enhancing the overall experience and ensure that the game achieves both its educational and entertainment objectives.

Another issue emerging during playtesting was the issue of question difficulty levels for some of the questions. While the questions were designed to promote critical thinking, they were found to be too difficult for experts. To solve this, we have decided to include multiple sets of questions of different difficulties. Also, we have added a description for each set. At the same time, the game remains difficult for high-level players by offering complex scenarios and more in-depth questions. The multiple-level approach in the questions makes it possible to tackle a wide range of skill levels, and therefore the game is attractive to a wide range of consumers.

Lastly, the criticism of the Eco-Reward cards and how they work helped us simplify the system for reward distribution. Awarding the reward to the member who gave the best answer helps eliminate inconsistencies within the group decision-making process and makes the contest fairer and more engaging. This change combined with the simplified movement mechanics will help make the game smoother and more enjoyable.

X. Future Work

Despite the improvements achieved through playtesting and feedback integration, there are several areas where the game can be enhanced, and more development can be carried out.

Currently, the game covers a broad range of sustainability challenges, but can go deeper into some aspects of green software development. For example, we can introduce more specific questions about energy-efficient algorithms, software architecture choices and the environmental impact of different programming languages. This will provide more learning opportunities, especially for advanced players.

To increase the game's flexibility, we could create other forms of play. For example, a "cooperative mode" in which teachers and students cooperate to achieve tasks could promote cooperation and cooperative problem-solving. This could be particularly useful in instructional settings like classrooms where cooperation and collaborative learning are often emphasized.

While there is a certain charm to the physical board game, we are considering making a computer version of the game. Then the game could be played from anywhere, allowing more players and including computer mechanics such as interactive tutorial levels or computer tracking of sustainability points. An online computer game would also make updating and expanding the game content easier, keeping the game up-to-date as new concepts in sustainable software engineering evolve.

By releasing the game open-source on GitHub, we invite users to provide new questions, suggestions for improvements, and other feedback. Such a user-driven development route can lead to the creation of new study material. We might even give players the option to tailor the questions of the game to be centered on topics most relevant to their personal learning requirements.

Based on the feedback regarding tile distribution, we will further work to balance the game. For example, the frequency and impact of rewards such as the double roll of dice will be adjusted so as not to adversely affect the competitiveness of the game. Similarly, the ratios of different tiles (e.g., Green Energy Question, Energy Black Hole, Eco-Reward) will also be looked at to discover a more balanced ratio between reward and challenge.

In the future, we will try to increase the exposure of the game by collaborating with schools, green groups, and coding camps. By sharing the game more widely through workshops, websites, and educational toolkits, we can encourage awareness of the importance of sustainable coding practice.

XI. Conclusion

The Sustainable Sprint game, inspired from "The Game of Goose", was created as an interactive learning experience meant to raise the awareness on sustainable software practice. Through the application of an enjoyable gameplay and questions that will stimulate thinking, this game is utilized to introduce the players to the ideas in energy efficient coding and green software development. We have enhanced the game mechanics, rules, and content to provide a more balanced and enjoyable experience with the iterative design, playtesting, and integrating feedback.

The playtesting has made sure that the game effectively encourages the critical reasoning and argumentation about the sustainability in software development. Participants really responded positively to the challenge based learning experience. Many people expressed that they have gained new information on green coding methodologies. However, areas for improvement were also observed, such as the reward balance, the tile distribution, and the question difficulty. With these adjustments we have developed a more fluid and pleasant experience while maintaining the game's educational value.

To guarantee the maximum accessibility and reach, we will release the game for free as an open-source project on GitHub. This will allow teachers, professionals, and other people to adapt and use the game in different contexts, such as classrooms, workshops, and workplaces. Moreover, we will also welcome any community contributions to expand the game content and improve the game in the long term.

In the future, we plan to try and make other game modes, even digital releases, and more distribution to further expand the reach and influence of this game. Through the ongoing improvement and community involvements, the Sustainable Sprint game can be a valuable learning tool that promotes the sustainable thinking within the software community. By integrating the sustainability into computer programming education in a fun and interactive way, we try to seek to motivate more programmers to adopt green coding practices and contribute towards a greener virtual world.

References

[1] Sedef Akinli Kocak. Green software development and design for environmental sustainability. 10 2013.

[2] Ilaria Caponetto, Jeffrey Earp, Michela Ott, et al. Gamification and education: A literature review. In *European conference on games based learning*, volume 1, page 50, 2014.

[3] Luca Caviglione, Mauro Gaggero, Enrico Cambiaso, and Maurizio Aiello. Measuring the energy consumption of cyber security. *IEEE Communications Magazine*, 55(7):58–63, 2017.

[4] Kaushik Dutta and Debra Vandermeer. Caching to reduce mobile app energy consumption. *ACM Trans. Web*, 12(1), September 2017.

[5] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *2014 47th Hawaii international conference on system sciences*, pages 3025–3034. Ieee, 2014.

[6] Wenlin Han, Minxian Xu, and Yanming Shen. Green cloud computing schemes based on networks: A survey. In *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, pages 2436–2440. Atlantis Press, 2013.

[7] Daniel Johnson, Ella Horton, Rory Mulcahy, and Marcus Foth. Gamification and serious games within the domain of domestic energy consumption: A systematic review. *Renewable and Sustainable Energy Reviews*, 73:249–264, 2017.

[8] Liang Luo, Wenjun Wu, Dichen Di, Fei Zhang, Yizhou Yan, and Yaokuan Mao. A resource scheduling algorithm of cloud computing based on energy efficient optimization methods. In *2012 International Green Computing Conference (IGCC)*, pages 1–6, 2012.

[9] Andrei Mccall. Sustainable computing and green information systems. 12 2024.

[10] Junya Michanan, Rinku Dewri, and Matthew J. Rutherford. Predicting data structures for energy efficient computing. In *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, pages 1–8, 2015.

[11] M. Mohankumar and M. Anand Kumar. An empirical study on green and sustainable software engineering. In *Advances in Software Engineering and Systems*, pages 95–99. World Scientific and Engineering Academy and Society (WSEAS), 2015.

[12] Ana Moreira, Ola Leifler, Stefanie Betz, Ian Brooks, Rafael Capilla, Vlad Constantin Coroama, Leticia Duboc, Joao Paulo Fernandes, Rogardt Heldal, Patricia Lago, Ngoc-Thanh Nguyen, Shola Oyedeji, Birgit Penzenstadler, Anne Kathrin Peters, Jari Porras, and Colin C. Venters. A road less travelled and beyond: Towards a roadmap for integrating sustainability into computing education, 2024.

[13] Gabriel Călin Nistor and Adrian Iacob. The advantages of gamification and game-based learning and their benefits in the development of education. In *The international scientific conference eLearning and software for education*, volume 1, pages 308–312. " Carol I" National Defence University, 2018.

[14] Artur F. S. Veloso, José V. R. Júnior, Matheus M. do N. Costa, Ricardo A. L. Rabelo, and

Placido R. Pinheiro. *A Comparative Evaluation of Monolithic and Microservice Architectures for Load Profiling Services in Smart Grids*, pages 17–36. Springer Nature Switzerland, Cham, 2025.

[15] Chi Xu, Ziyang Zhao, Haiyang Wang, and Jiangchuan Liu. On the interplay between network traffic and energy consumption in virtualized environment: An empirical study. In *2014 IEEE 7th International Conference on Cloud Computing*, pages 392–399, 2014.

[16] Qingbo Zhu, F.M. David, C.F. Devaraj, Zhenmin Li, Yuanyuan Zhou, and Pei Cao. Reducing energy consumption of disk storage using power-aware cache management. In *10th International Symposium on High Performance Computer Architecture (HPCA'04)*, pages 118–118, 2004.